Journal of Information Systems Education, Vol. 14(1)

Teaching Tip

Utilizing Simple Hacking Techniques to Teach System Security and Hacker Identification

Aaron D. Sanders
Clarion University
Clarion, Pennsylvania 16214
ads7971@rit.edu

ABSTRACT

This first half of this paper details the tools and methodologies employed to determine the identity and physical location of a hacker who infiltrated a server and altered a Web page. The second half of this paper recreates the scenario in a laboratory environment, in order to instruct students on system administration, server security, network management, and basic data communications.

Keywords: Hands-on exercises, system administration, IIS security, server configuration

1. INTRODUCTION

As an undergraduate student working on my Bachelor of Science degree in Information Systems, I had the opportunity to work as a part-time Network Administrator. My duties included performing administrative and troubleshooting tasks on the servers, workstations, and network. The domain consisted of five servers, all running Windows 2000 Server and Internet Information Services (IIS) 5.0.

One morning I received an email message informing me that one of the servers had been compromised, and one of the Web pages altered. The actual damage was minor: The page had been changed from its original state to a black background with a large yellow smiley face, and the message "You've been HaCkEd. Have a nice day!" Although no malignant damage had occurred, I realized that this provided a unique opportunity for me to attempt to determine the identity of the hacker (I choose to use the more popular although incorrect term "hacker", because it causes less confusion than "cracker", which is the correct term in this situation). Not only could I examine the tools and methodologies employed in similar scenarios, but this situation would also provide valuable knowledge in server security and system administration.

2. PROCEDURE

The first step in attempting to determine the identity of the hacker was to search for clues in the directory containing the altered page. Interestingly enough, the process methodology mirrored standard troubleshooting procedures, or a police officer attempting to solve a crime. One must start at the scene of the crime and gather as many direct clues as possible, then work their way outward, examining the larger picture. Since the directory contained published Web pages, it existed as a subdirectory to the wwwroot directory (\inetpub\wwwroot). The first piece of useful information that appeared was the created and modified dates for the altered Hypertext Markup Language (HTML) file. Although these dates were not an absolute fingerprint, they provided a frame of reference to use in searching for other clues.

The next step was the examination of the _vti_cnf subdirectory, which on servers with FrontPage Extensions enabled, is used by FrontPage to store configuration information for files in the parent directory. Every directory available via the Web will have a _vti_cnf subdirectory, which contains configuration files for each HTML file in the parent directory. These configuration files will have the same filename and extension as their HTML counterparts, with the only difference being that when you view the configuration files in a browser, configuration information will be displayed, rather than the actual page they mirror. The configuration files residing in the _vti_cnf subdirectory contain some important information, including the file's author, last time

modified, next to last time modified, and time created. (On a configuration note, it is highly recommended that the computer under examination be configured to view hidden files, as some of the files in the _vti_cnf directory may be hidden files). As they logically should, the dates and times listed in the configuration file for the altered page matched the dates and times discovered by viewing the properties on the altered page. The author's name was the most important piece of information gained from the configuration file, and a crucial piece of the puzzle. The name was a valid account in the Active Directory for the server, and someone who would not be hacking servers and altering other people's Web pages. At this point, the evidence seemed to suggest that someone had correctly guessed the password of a user that had access to the server and directory, connected to the directory, and altered the Web page.

The next step was to examine the various log files created by IIS and other connectivity programs. The first log files requiring examination are the Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP) logs that IIS automatically creates. The default storage location for the HTTP log files is the \%windir%\System32\LogFiles\W3SVC1 (%windir% being the directory that Windows 2000 is installed in, normally \WINNT) directory, their default filename is ex*LOGDATE*.log, and there is one file for each day that IIS had HTTP activity. Although the filenames and location can vary depending on options chosen during IIS configuration, it is rare for the defaults to be changed. If finding their location is problematic, the Start Menu's Find option can be used to search for files named "*.log" on all drives. This will reveal their location and naming convention. The HTTP logs contain GET (when someone requests a file from the server via the Web) requests and POST (when someone places a file on the server via the Web) requests, although the HTTP logs record POST requests only when files are placed on the server using Microsoft FrontPage or similar methods. The log files contain the date of request and files requested, along with the Internet Protocol (IP) address that made the request.

In order to gather information from the established frame of reference, the HTTP logs corresponding to the week before and week of the created and modified dates for the changed version of the page were examined, and a peculiar trend was discovered. GET requests for the altered page were rare under normal circumstances, yet the same IP address repeatedly viewed the page during those two weeks. On a daily basis the logs showed multiple GET requests logged for the page from the same IP address. The IP address in question was using a "ten-dot" or private IP addressing scheme (10.x.x.x), which was the addressing scheme used for sections of the Local Area Network (LAN). The examination of the log files continued until the necessary entry was discovered: A POST request by the IP address in question placing the changed Web page into the directory on the server, complete with the creation of the configuration files. At this point, it appeared that the hacker had used FrontPage on a computer somewhere on the LAN to upload the page to the server, gaining access by correctly guessing the password of one of the users with rights to the server and directory.

A thorough search also requires the examination of the log files that IIS creates when someone connects (or attempts to connect) to the server via FTP. The default storage location for the FTP log files is the \%windir%\System32\LogFiles\MSFTPSVC1 directory, their default filename is ex*LOGDATE*.log, and there is one file for each day that IIS had FTP activity. As with the HTTP logs, the default location and naming convention can be altered, and the Find option can be used to uncover their location and naming convention if the defaults were not used. The FTP logs contain the IP address and username of individuals that attempted to connect to the server via FTP, and the outcome of their attempt (successful or unsuccessful connection). If the attempt was successful, the FTP logs will show the files that the user copied to or deleted from the server. The FTP logs files contained startling (although unrelated) information: On a daily basis (and often multiple times per day), anonymous individuals were attempting to gain access to the server via FTP. They attempted to connect using the anonymous, guest, and Internet User (IUSR_*computername*) default accounts created by IIS. The FTP log files for those two weeks showed no signs of activity by the IP address in question and no signs of successful intrusion by any other IP address. The FTP log files also provided some interesting information regarding which authorized individuals connect the most via FTP. This information was useful because it revealed which individuals most utilized the available resources. In addition, when monitoring for abnormal activity, it is beneficial to know the type and level of normal activity.

Examining the log files proved that an important piece of knowledge is that of the applications currently utilized in the public production environment. Possessing knowledge of the popular HTTP and FTP clients and servers and HTML editors can be crucial in apprehending a hacker. Programs that allow remote connectivity create (or can be configured to create) log files, and knowing the location of the log files, the information they contain, and how to utilize them can be very helpful. In addition, it is often useful to check the log files for any intermediate devices, such as routers or firewalls.

The next objective was to determine the physical location of the hacker. This posed a rather large problem because IP address assignments do not correlate to physical locations. The evidence seemed to show that the hacker was a member of the LAN, since the IP address in question followed the private IP addressing scheme used for portions of the LAN, and since private IP addresses are not natively routable, the IP address

probably could not have originated outside the LAN.

VisualRoute 6.0b was the tool used to determine the physical location of the hacker. VisualRoute is a **ping**, **whois**, and **tracert** program that displays the results of its tests in a table and on a world map. VisualRoute can trace the route to an IP address or domain name, and return the IP address, node name (including registrant, administrative contact, address, phone number, when the domain lease ends, and IP addresses of the domain's DNS servers), worldwide geographic location (including latitude and longitude), time zone, elapsed response time in milliseconds, and network provider (including IP address block, DNS server IP addresses, mailing address, and telephone number). VisualRoute returns this information for every hop along the way, only failing at the end, if the destination network is set to block the Internet Control Message Protocol (ICMP) packets used by the **ping** command. Organizations and individuals often configure firewalls to block external ICMP packets from reaching their internal network. In this situation, VisualRoute can still reveal pertinent information about the end node (for example the server housing a Web Site), but it cannot reveal the number of hops from the edge of the internal network to the specific node. VisualRoute displays all of the information it gathers in a table in the top half of the screen, and on the bottom half of the screen it plots every physical location on a world map, and connects them to show the physical route. Visual Route also displays the server protocol type (HTTP, FTP, et cetera), Web Server software brand (Microsoft IIS, Apache, et cetera) and version. VisualRoute was given the IP address of the hacker's computer in an attempt to gain more information about the hacker's physical location. As expected, VisualRoute traced the IP address to the LAN. The VisualRoute trace revealed the identity of the hacker, as he had foolishly decided to perform the hack from a computer that used his given name for its computer name. VisualRoute plainly displayed the computer name in the "node name" field.

In an attempt to prevent further intrusions, Microsoft Baseline Security Analyzer (MBSA) was utilized to analyze current security settings. MBSA (formerly known as the Microsoft Personal Security Advisor or PSA), is a program available from Microsoft's Web Site that runs on Windows 2000 and Windows XP systems. MBSA can scan for missing hotfixes and vulnerabilities in the following products: Windows NT 4.0 Server and Workstation, Windows 2000 Server and Professional, Windows XP, IIS 4.0 and 5.0, SQL Server 7.0 and 2000, Internet Explorer (IE) 5.01 and later, and Office 2000 and XP. MBSA creates and stores individual Extensible Markup Language (XML) security reports for each computer scanned and displays the reports in HTML form in the program's graphical user interface. MBSA will check the status of the guest account, suggest renaming the administrator account, list all users that have administrative rights, check for weak passwords, and determine if passwords are set to expire. In addition, MBSA will determine the file system of each disk drive, suggest enabling auditing, detect unnecessary services, display all shared directories, and check the status of all patches and hotfixes. MBSA was used to analyze the hacked server, and it determined that multiple users had weak passwords, including the user whose account had been used to alter the Web page.

Another useful tool (although one that was not necessary for this operation) is a packet sniffer. A packet sniffer is one possible tool to use in determining the IP address of an unknown party. Ufasoft Sniffer XP 3.5 is a powerful packet sniffer that can monitor Address Resolution Protocol (ARP) packets, Ethernet packets, ICMP packets, IP packets, Transmission Control Protocol (TCP) packets and connections, and User Datagram Protocol (UDP) packets. For every captured packet it gives hexadecimal and text interpretation of the source and destination Media Access Control (MAC) addresses, source and destination IP addresses, IP data, TCP source and destination port numbers and port type, TCP sequence number, TCP acknowledgement number, TCP window number, and TCP data. Ufasoft Sniffer is a good choice because it is easy to use, and displays the information side by side on the screen in hexadecimal and text. Clicking on a text field in the left pane (for example source IP address) highlights the same information in the hexadecimal portion in the right pane. This feature makes Ufasoft Sniffer a useful tool for teaching students about packet composition and showing them how and where information is contained, regardless of their knowledge of the hexadecimal system. Examining the composition of an actual captured packet allows students to gain better understanding of packet standards then simply viewing packet diagrams in a textbook. It is always interesting the first time one examines unencrypted email messages or Web pages, and realizes how easily readable the data is with a packet sniffer.

3. EDUCATIONAL IMPORTANCE

Crucial skills for today's Information Technology (IT) professional include the ability to secure networks and servers, and to detect, determine the source of, and correct problems. Server security is paramount in the modern information society, and news stories of high profile hacks are becoming more common. The demand for professionals with strong security skills is growing, and colleges across the nation have begun adding undergraduate and graduate programs in electronic information security. The skills and methodologies detailed in this paper are crucial for the knowledgeable student, and would fit well into a LAN or system administration class, or any other class where system security is concerned.

The cost of creating a laboratory environment to employ hacking exercises can be minimal, with the most

problematic items being space to set up the laboratory environment, and the required computers. A free trial of Windows 2000 Server is available with a one hundred twenty day limit, and the software can run on a 166Mhz Pentium 1 computer with only 64MB of Random Access Memory (RAM). No client computers are required if the hack is going to be performed from a remote location. The laboratory requires only one client machine if the hack is going to be performed locally, although more could be used if the budget allows, and would be useful if the laboratory is being used to demonstrate multiple topics. Although the clients can run any operating system, they must be running Windows 2000 or Windows XP if the instructor desires to run the MBSA from the client machines to test client and server security. VisualRoute has a thirty-day evaluation period, the cost for one license is $39.95, and VisualWare offers license packages up to two hundred and fifty users. Microsoft's MBSA is available free from Microsoft's TechNet Web Site. The Ufasoft Sniffer has a thirty-day trial, and costs $29 for a one-user license. The ease of use and available features vary greatly between packet analyzer programs, and it is highly recommended that the instructor evaluate a variety of products before deciding on one to use.

The major steps required to create the laboratory are to install and configure Windows 2000 Server, install and configure IIS, create subdirectories in the wwwroot directory with Web pages in them, create virtual directories for those directories, create user accounts with FTP rights, and give those FTP users rights to the directories. Detailed instructions for each step are readily available on the Web and in many published books. The instructor can use any preferred method to secretly connect to the server from one of the client machines (or anywhere in the world if the server has outside access), and change as many Web pages as desired. It is then left to the students to attempt to determine how the server was compromised, how the Web pages were changed, and as much information about the hacker's identity and physical location as possible.

Hacker identification exercises are a good technique for allowing the instructor to discuss the tools and methodologies used in such an operation, and the theory behind server security. This provides a segue to discuss topics such as disabling or carefully administering the guest and IUSR_*computername* accounts, changing the name of the administrator account, monitoring and controlling the number of users that receive administrative rights, monitoring and controlling user password strength, deleting old accounts for users that no longer exist, and disabling anonymous FTP. This branches into other areas of system administration, such as setting password expiration, length, and character rules when creating new users. It also provides an avenue to discuss server and network baselining, a very important management aspect of system administration. Baselining is important because it determines performance standards that can be useful in detecting a network problem. The majority of hacked servers become FTP servers for illegal distribution of software and music ("Warez"). If this happens, the performance of the hacked server and the network it resides on will greatly decrease. Comparing the current performace of the network to the established baselines uncovers problems. Networked organizations can use network monitoring software to constantly monitor the network and alert an administrator by pager or email if certain thresholds are reached.

4. SOCIAL RESPONSIBILITY

The techniques discussed in this paper could have positive or negative ramifications, and require an ethics discussion. Uncertainty exists among security professionals regarding the danger of using hacking skills to improve a student's security knowledge. The main concern voiced by security professionals is that teaching hacking techniques in the classroom could enable a student to improve their skills from the level of "script kiddie" (an individual who possesses no substantial knowledge of their own, instead executing hacks using automated tools that others have created) into a knowledgeable (and potentially dangerous) hacker. A student that chooses the ethical path can become a security expert or highly skilled system administrator. A student that chooses the unethical path could be part of a large wave of skilled hackers, wreaking havoc on the Web. It is crucial to exercise caution when endowing an individual or organization with potentially dangerous knowledge and tools.

The majority of security experts agree that difficulty exists in teaching security and system administration, without minimally talking about popular vulnerabilities, and the methods used to correct them. In order to fortify a given system, a security professional must know the security weaknesses that can occur on the system. Most experts believe that courses should use hacking techniques only when the students are reputable individuals from established organizations. Few security experts support teaching hacking skills to college-aged students or younger. Many commercial organizations, such as Internet Security Systems (ISS), provide security and vulnerability testing to organizations, along with ethical hacking courses. ISS only admits students to a course if the student can prove they are from a reputable, established company, and if the student can prove their own integrity. ISS has a positive reputation to uphold, and the company has no desire in training an army of malicious hackers.

The importance of ethical discussions in higher education programs is increasing, and colleges and universities are adding ethics requirements to their degree programs. A course that teaches security or system administration using hacking techniques could

fill a portion of a student's ethics requirement, if the course includes the proper ethics discussions. It is crucial in the academic environment to push ethical and legal issues, because if academia does not, few others will. Emphasizing the importance of the topic is difficult, and simply adding discussions on ethics into courses does not guarantee that a course will produce ethical computer experts. If simply teaching students about ethics created ethical people, then society would be free of theft and murder.

The risk of training and equipping a future hacker is not great enough to forgo properly teaching security. Higher education programs around the world are adding security courses and degrees, and academia is the best place for teaching the necessary skills, along with the accompanying ethics. Resources such as the Hackademy in Paris, the Astalavista Security Group's Web Site, "2600: The Hacker Quarterly", and the DEFCON hacker's convention, will provide the necessary skills and information, with or without adding ethics. It is crucial for academia to use its time with students to provide the ethical and legal lessons that other sources may not provide.

5. CONCLUSION

One of the most important elements of a strong computing program is hands-on application of theoretical knowledge. Students often have difficulty understanding how the pieces fit together, and cannot visualize how topic discussions relate to real-world application without hand-on exercises to reinforce their learning. It is important to provide as much hands-on learning as possible, without sacrificing too much of the necessary theory elements.

The methodologies and tools used to identify a hacker build a strong knowledgebase in students, and reinforce similar topics in system administration, server security, network management, and data communications. Recreating these exercises in a laboratory environment provides a cost effective method of instructing students with crucial tools and techniques. This can build students' general problem solving ability: Faced with a problem they may not know how to solve, students must determine the best methods and tools to solve the problem. Upon first notification of the hacked server, I had no prior knowledge of the _vti_cnf directory or its use, what logs IIS created and where it stored them, and I had never used VisualRoute for anything other than a toy. When presented with this problem, I was able to think logically through the process of finding the solution, and realized that the tools I needed were available. Cognitive thinking abilities are one of the most important attributes for a student to posses, because in the real world, problems do not always occur exactly as the textbook described, and the answer is not available in the back of the book. Challenging students to solve a new problem by combining existing knowledge with knowledge gained through specific research will benefit them throughout their career.

6. REFERENCES

Anonymous. *Maximum Security: A Hacker's Guide to Protecting Your Internet Site and Network.* Indianapolis: Sams Publishing, 1998.

Carpenter, Susan. "Where Hackers Teach the Art of Self-Defense." Los Angeles Times 2002: <http://www.ttivanguard.com/bru_reconn/hackademy.pdf>.

Matthews, Martin S., and Erik B. Poulsen. *FrontPage 2000: The Complete Reference*. Berkeley: Osborne, 1999.

Microsoft Baseline Security Analyzer. Microsoft Corporation. <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/Tools/mbsahome.asp>.

Norfolk, David. "Understanding Ethical Hacking." PC Network Advisor Mar. 2001. <http://www.itp-journals.com/nasample/M04133.pdf>.

Schneier, Bruce. *Secrets and Lies: Digital Security in a Networked World.* New York: John Wiley & Sons, 2000.

"Should Ethical Hacking be Taught as a Career Course?" Express IT People April 2002. <http://www.itpeopleindia.com/20020422/cover1.shtml>.

Ufasoft Sniffer. Ufasoft Company. <http://www.ufasoft.com/sniffer>.

VisualRoute by Visualware. Visualware Incorporated. <http://www.visualware.com/visualroute/index.html>.

AUTHOR BIOGRAPHY

Aaron D. Sanders is a student in the Master of Science in Information Technology (Networking and Telecommunications Concentrations) program at Rochester Institute of Technology. He received his Bachelor of Science in Information Systems from Clarion University of Pennsylvania. Upon completion of his M.S., he plans to pursue a career in computing education. This paper is the first in a series concerning information policy, security, and ethics, which will cumulate with his Masters Thesis.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096